

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-149280

(43)Date of publication of application : 02.06.1998

(51)Int.Cl.

G06F 9/06

(21)Application number : 09-249098

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 12.09.1997

(72)Inventor : SHIMADA TAKESHI  
 MIZUTANI HIROYUKI  
 SAKURAI SHIGEAKI  
 GOTO YUKIHIRO  
 TSUKADA HIROSHI  
 GOTO KAZUHISA  
 ENDO KYOICHI  
 ONO AKIRA

(30)Priority

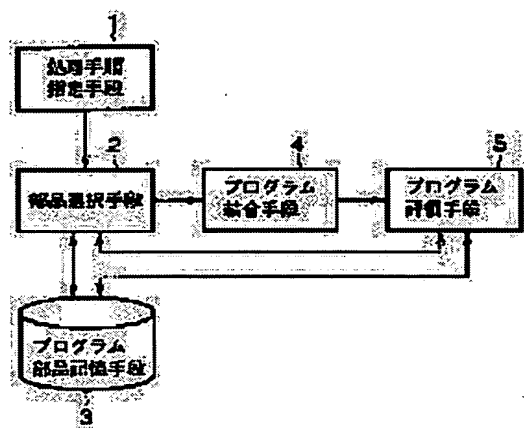
Priority number : 08245359 Priority date : 17.09.1996 Priority country : JP

## (54) DEVICE AND METHOD FOR PROGRAM COMPOSITION

(57)Abstract:

PROBLEM TO BE SOLVED: To limit choices of program components to some of them by using an outline processing procedure, and to compose a desired program with efficiency by selecting program components corresponding to the outline processing procedure out of stored program components, combining the program components, and evaluating the performance of the composed program.

SOLUTION: A processing procedure specifying means 1 specifies the outline processing procedure that the desired program should follow. A program component storage means 3 holds the program components by component classifications. A component selecting means 2 replaces individual outline processes included in the outline processing procedure with the program components selected out of the program component storage means 3. A program combining means 4 combines the program components selected by the component selecting means 2 to compose a program. A program evaluating means 5 determines an evaluated value of the said composed program.



## LEGAL STATUS

[Date of request for examination]	29.08.2000
[Date of sending the examiner's decision of rejection]	01.06.2004
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	3601949
[Date of registration]	01.10.2004
[Number of appeal against examiner's decision of	2004-13531

rejection]

[Date of requesting appeal against examiner's decision of 01.07.2004

rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-149280

(43) 公開日 平成10年(1998) 6月2日

(51) Int.Cl.<sup>6</sup>

G 0 6 F 9/06

識別記号

5 3 0

F I

G 0 6 F 9/06

5 3 0 X

5 3 0 W

審査請求 未請求 請求項の数24 O L (全 24 頁)

(21) 出願番号 特願平9-249098

(22) 出願日 平成9年(1997) 9月12日

(31) 優先権主張番号 特願平8-245359

(32) 優先日 平8(1996) 9月17日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72) 発明者 島田 毅

神奈川県川崎市幸区柳町70番地 株式会社

東芝柳町工場内

(72) 発明者 水谷 博之

神奈川県川崎市幸区柳町70番地 株式会社

東芝柳町工場内

(72) 発明者 櫻井 茂明

神奈川県川崎市幸区柳町70番地 株式会社

東芝柳町工場内

(74) 代理人 弁理士 鈴江 武彦 (外6名)

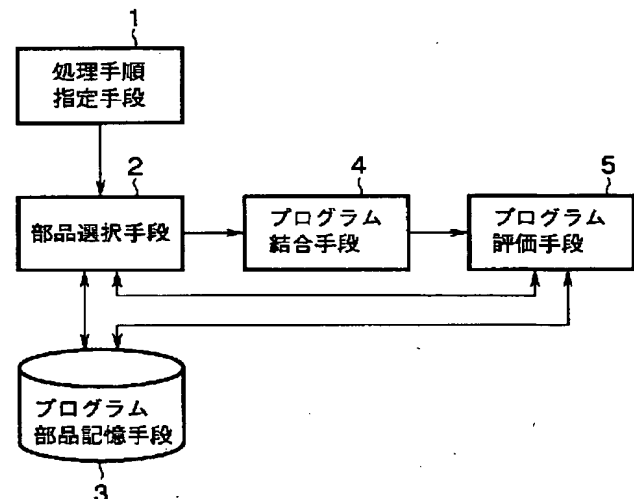
最終頁に続く

(54) 【発明の名称】 プログラム合成装置及びプログラム合成方法

(57) 【要約】

【課題】 概略処理手順を使って、プログラム部品の選択肢を一部分に限定することにより、効率よく（短時間に容易に）所望のプログラムを合成できるプログラム合成方法およびそれを用いたプログラム合成装置を提供することを目的とする。

【解決手段】 予め分類されたプログラム部品を記憶する部品記憶手段（3）と、前記プログラム部品の分類に基づき所望のプログラムの概略処理手順を指定する処理手順指定手段（1）と、前記部品記憶手段に記憶されたプログラム部品から前記概略処理手順に対応するプログラム部品を選択する部品選択手段（2）と、この部品選択手段で選択されたプログラム部品を結合してプログラムを合成する合成手段（4）と、この合成手段で合成されたプログラムの性能を評価する評価手段（5）とを具備し、前記部品選択手段は、前記評価手段で得られた評価値を参照してプログラム部品を選択する。



**【特許請求の範囲】**

**【請求項1】** プログラム部品を組み合わせる所望のプログラムを合成するプログラム合成装置において、予め分類されたプログラム部品を記憶する部品記憶手段と、前記プログラム部品の分類に基づき所望のプログラムの概略処理手順を指定する処理手順指定手段と、前記部品記憶手段に記憶されたプログラム部品から前記概略処理手順に対応するプログラム部品を選択する部品選択手段と、前記部品選択手段で選択されたプログラム部品を結合してプログラムを合成する合成手段と、前記合成手段で合成されたプログラムの性能を評価する評価手段と、を具備することを特徴とするプログラム合成装置。

**【請求項2】** 前記部品選択手段は、前記評価手段で得られた評価値を参照してプログラム部品を選択することを特徴とする請求項1に記載のプログラム合成装置。

**【請求項3】** 前記評価手段でプログラムを評価する際の評価尺度を定義する手段をさらに具備することを特徴とする請求項1に記載のプログラム合成装置。

**【請求項4】** プログラム部品を組み合わせる所望のプログラムを合成するプログラム合成装置において、予め分類されたプログラム部品を記憶する部品記憶手段と、予め指定された条件に従って、前記プログラム部品の分類に基づき所望のプログラムの概略処理手順を生成する処理手順生成手段と、前記部品記憶手段に記憶されたプログラム部品から前記概略処理手順に対応するプログラム部品を選択する部品選択手段と、前記部品選択手段で選択されたプログラム部品を結合してプログラムを合成する合成手段と、前記合成手段で合成されたプログラムの性能を評価する評価手段と、を具備することを特徴とするプログラム合成装置。

**【請求項5】** 前記処理手順生成手段は、前記評価手段で得られた評価値を参照して概略処理手順を生成し、前記部品選択手段は前記評価値を参照してプログラム部品を選択することを特徴とする請求項4に記載のプログラム合成装置。

**【請求項6】** 前記評価手段でプログラムを評価する際の評価尺度を定義する手段をさらに具備することを特徴とする請求項4に記載のプログラム合成装置。

**【請求項7】** 前記プログラム部品の分類に基づき所望のプログラムの概略処理手順の全部あるいは一部を指定する処理手順指定手段をさらに具備することを特徴とする請求項4に記載のプログラム合成装置。

**【請求項8】** プログラム部品を組み合わせる所望のプログラムを合成するプログラム合成装置において、

予め分類されたプログラム部品を記憶する部品記憶手段と、前記プログラム部品の分類に基づき所望のプログラムの概略処理手順を指定する処理手順指定手段と、前記部品記憶手段に記憶されたプログラム部品から前記概略処理手順に対応するプログラム部品を、遺伝的アルゴリズムに基づいて探索する部品探索手段と、前記部品探索で探索されたプログラム部品を結合してプログラムを合成する合成手段と、前記合成手段で合成されたプログラムの性能を評価する評価手段と、を具備することを特徴とするプログラム合成装置。

**【請求項9】** 前記部品探索手段は、前記処理手順指定手段より与えられる概略処理手順に含まれる個々の概略処理の並びに従って、前記プログラム部品記憶手段からプログラム部品を選択して配列し、染色体を得ることを特徴とする請求項8に記載のプログラム合成装置。

**【請求項10】** 前記染色体は、実行順序を持った一連のプログラム部品の流れであり、前記部品探索手段により異なった配列方法で作られる複数の染色体は、染色体集団を構成することを特徴とする請求項9に記載のプログラム合成装置。

**【請求項11】** 前記染色体集団は、前記プログラム結合手段に供給され、当該プログラム結合手段は、個々の染色体についてプログラム部品間の引数対応を決定することを特徴とする請求項10に記載のプログラム合成装置。

**【請求項12】** 前記プログラム結合手段により引数対応を付られた染色体集団はプログラム評価手段に供給され、当該プログラム評価手段は、個々の染色体をプログラムとして実行し、各々のプログラムの評価値を算出することを特徴とする請求項11に記載のプログラム合成装置。

**【請求項13】** 前記部品探索手段は、より良い評価値を得る可能性の高い染色体を交叉又は突然変異を用いて生成し直すことを特徴とする請求項9に記載のプログラム合成装置。

**【請求項14】** 予め分類された複数のプログラム部品のうち所定のプログラム部品を組み合わせる所望のプログラムを合成するプログラム合成方法において、前記プログラム部品の分類に基づき所望のプログラムの概略処理手順を指定する指定ステップと、前記複数のプログラム部品から前記概略処理手順に対応するプログラム部品を選択する部品選択ステップと、前記部品選択ステップにおいて選択されたプログラム部品を結合してプログラムを合成する合成ステップと、前記合成ステップにおいて合成されたプログラムの性能を評価して評価値を求め、その評価値に基づいて前記部品選択ステップ、合成ステップを繰り返す評価ステップと、

を具備することを特徴とするプログラム合成方法。

【請求項15】 前記部品選択ステップは、前記評価ステップで得られた評価値を参照してプログラム部品を選択することを特徴とする請求項14に記載のプログラム合成方法。

【請求項16】 予め分類された複数のプログラム部品のうち所定のプログラム部品を組み合わせることで所望のプログラムを合成するプログラム合成方法において、所望のプログラムの概略処理手順を所定の規則に従って生成する概略処理手順生成ステップと、前記複数のプログラム部品から前記概略処理手順に対応するプログラム部品を選択する部品選択ステップと、前記部品選択ステップにおいて選択されたプログラム部品を結合してプログラムを合成する合成ステップと、前記合成ステップにおいて合成されたプログラムの性能を評価して表価値を求め、その表価値に基づいて前記部品選択ステップ、合成ステップを繰り返す評価ステップと、を具備することを特徴とするプログラム合成方法。

【請求項17】 前記処理手順生成ステップは、前記評価ステップで得られた評価値を参照して概略処理手順を生成し、前記部品選択ステップは前記評価値を参照してプログラム部品を選択することを特徴とする請求項16に記載のプログラム合成方法。

【請求項18】 予め分類された複数のプログラム部品のうち、所定のプログラム部品を組み合わせることで所望のプログラムを合成するプログラム合成方法において、前記プログラム部品の分類に基づき所望のプログラムの概略処理手順を指定する指定ステップと、前記分類された複数のプログラム部品から前記概略処理手順に対応するプログラム部品を、遺伝的アルゴリズムに基づいて探索する部品探索ステップと、この部品探索で探索されたプログラム部品を結合してプログラムを合成する合成ステップと、前記合成ステップにおいて合成されたプログラムの性能を評価して表価値を求め、その表価値に基づいて前記部品選択ステップ、合成ステップを繰り返す評価ステップと、を具備することを特徴とするプログラム合成方法。

【請求項19】 前記部品探索ステップは、前記処理手順指定ステップにおいて指定された概略処理手順に含まれる個々の概略処理の並びに従って、前記分類された複数のプログラム部品からプログラム部品を選択して配列し、染色体を得ることを特徴とする請求項18に記載のプログラム合成方法。

【請求項20】 前記染色体は、実行順序を持った一連のプログラム部品の流れであり、前記部品探索ステップによって異なった配列方法で作られる複数の染色体は、染色体集団を構成することを特徴とする請求項19に記載のプログラム合成方法。

【請求項21】 前記染色体集団は、前記プログラム結合ステップに供給され、当該プログラム結合ステップにおいて、個々の染色体についてプログラム部品間の引数対応が決定されることを特徴とする請求項20に記載のプログラム合成方法。

【請求項22】 前記プログラム結合ステップにおいて引数対応が付られた染色体集団は前記評価ステップに供給され、当該評価ステップは、個々の染色体がプログラムとして実行し、各々のプログラムの評価値を算出することを特徴とする請求項21に記載のプログラム合成方法。

【請求項23】 前記部品探索ステップは、より良い評価値を得る可能性の高い染色体を交叉又は突然変異を用いて生成し直すことを特徴とする請求項19に記載のプログラム合成方法。

【請求項24】 予め分類された複数のプログラム部品のうち所定のプログラム部品を組み合わせることで所望のプログラムを合成するプログラムコード手段が記録された、計算機により読み出し可能な記録媒体において、前記プログラム部品の分類に基づき所望のプログラムの概略処理手順を指定させるインストラクション手段と、前記複数のプログラム部品から前記概略処理手順に対応するプログラム部品を選択させるインストラクション手段と、前記選択されたプログラム部品を結合してプログラムを合成させるインストラクション手段と、前記合成されたプログラムの性能を評価して表価値を求め、その表価値に基づいて前記プログラム部品の選択、及びプログラム合成を繰り返させるインストラクション手段と、を具備することを特徴とする計算機により読み出し可能な記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピューターのプログラムを予め登録されているプログラム部品から合成する装置及び方法に関し、特に、最初は暫定的なプログラムを生成し、そのプログラムに対して徐々に改良を加えることによって性能の良い最終的なプログラムを合成するプログラム合成装置およびプログラム合成方法に関する。

【0002】

【従来の技術】過去に蓄積したプログラム部品を組み合わせることで新しいプログラムを作る技術、即ちプログラムの部品合成においては、処理する課題に関して人間の専門家が持っている知識を、機械で扱える形式に纏める必要がある。

【0003】画像処理プログラムを作る場合で説明すると、画像処理の方法や、処理の対象画像の光学的性質（たとえばパステル画のように全体がぼやけていると

か、直線的な部分が多いなど）、使用する画像処理サブルーチンの内容や入出力変数の型、サブルーチンの効果（そのサブルーチンを施すことによる画像の変化）などが、人間の専門家で持っている知識の例である。

【0004】このうち、画像処理サブルーチンの入出力引数の型のような形式的な知識は、マニュアルなどの文書化された形式に体系的にまとめられており、また内容も正確なので、知識獲得は容易である。

【0005】これに対し、専門家が沢山の選択肢から適切な画像処理サブルーチンを選ぶとき、画像上のどんな特徴に着目し、どんなルールに基づいて判断しているのか、といった複雑な知識は、画像処理分野の長年の研究にも関わらず、機械化できるような形式に体系的にまとめられてはいない。画像処理の優秀な専門家でも、自分がどんな論理的思考に基づいてプログラムを作成しているのかを明確に意識することは稀であり、本人ですら、自分の知識を体系立てて整理することは難しい。まして他人が、インタビューなどの方法で専門家から知識を獲得することは不可能であると言っても過言ではない。

【0006】過去、エキスパートシステムを、プログラムの部品合成に応用することが試みられてきたが、上述の困難さのため、内省ないしはインタビューによって熟練作業から獲得できるのは、表面的な知識に限定されてしまう。熟練者から引き出せる表面的な知識だけでは、実用に耐え得るプログラムを合成するには甚だ不足であった。

【0007】この不足を補う目的で行なわれたのは、熟練者の作業を外から観察し、熟練者が使っている知識を推論し、人工的に作ることであった。しかしこの方法には、以下の問題点がある。

【0008】すなわち、いろいろな処理対象に適用できるよう網羅して人工的な知識を作り上げるには、膨大な労力が必要になるという問題があり、現実には狭い適用範囲に限定した知識しか準備できず、これを利用した部品合成もまた、自ずから適用分野が限定されてしまう。

【0009】また人工的な知識は、熟練者が長年の経験によって蓄積した知識に比較して不正確にならざるを得ず、したがって合成したプログラムの性能も、熟練者が作成するプログラムに及ばない。

【0010】上述の問題に対する解決策として、暫定的なプログラムを出発点とし、修正を繰り返して加えることで、徐々に性能の良いプログラムを合成する技術が提案されている。「認識機構学習システムMIRACLE-IVにおける画像処理戦略の学習」（電子情報通信学会技術研究報告PRU88-20記載）に見られる、遺伝的アルゴリズムによる技術はこの例である。この技術は、試行錯誤的に部品を組み合わせでプログラムを作り、そのプログラムの処理結果と、処理結果の目標の差によってプログラムを評価し、評価値が所望の値に達するまで修正を繰り返してゆく。

【0011】この技術は、知識としてプログラム部品同士の接続可否の情報、つまり入出力引数の結合可否の情報という形式的な知識だけを使う。画像処理サブルーチンを選ぶルールのような形式化できない知識は必要としないので、専門家の複雑な知識の獲得という困難かつ複雑な作業を経ることなく、性能の良いプログラムを得られる。

【0012】しかし上述の従来技術には、以下の問題点がある。

【0013】第一の問題は、合成されるプログラムの性能を高めるには、できるだけ多くの種類のプログラム部品を用意することが肝要であるが、これは一方で、部品組合せの場合の数の爆発的な増大の原因になってしまうという点にある。組合せの数の爆発的な増大によって、必要な試行錯誤の回数が幾何級数的に増え、そのため、所望の性能のプログラムを得るまでに掛かる時間が、現実的範囲に収まらないことがある。このように、ここで述べた従来技術は、現実規模のプログラムの部品合成に適用することが困難であった。

【0014】また第二の問題は、合成するプログラムを実際の処理結果と目標とする処理結果がどのくらい一致するか（出力値と目標値の間の誤差の大きさや、出力値と目標値が完全に一致するデータの件数の全テストデータの件数に対する割合などで定義される）ということのみで評価するため、処理結果が全く同じになる複数のプログラムの優劣を評価できないという点にある。処理結果が全く同じでも、使用するプログラム部品の違いによって、合成したプログラムの実行に要する時間、実行に要するプログラムコードのサイズ、実行に要する記憶領域のサイズ、プログラムの構造の複雑さなどが変わる。またプログラム部品の信頼性、拡張性の良さ、等の主観的評価にもとづく評価も有り得る。これらのどの評価尺度に対してプログラムを最適化してゆくのがよいのかは課題によってまちまちである。

【0015】一例を挙げる。プログラムを実行するターゲットマシンが安価なマイコンの場合は主記憶装置の容量は一般に小さく、プログラムが消費する記憶領域の大きさは厳しく制限されるので、実行に要する記憶領域のサイズができるだけ少ないプログラムがより望ましい。一方、銀行のオンライン取り引き記録処理プログラムの場合、大規模な計算機上で処理されるため消費する記憶領域のサイズには制限がないものの、大量の取り引き情報を一定の時間内に捌くことが求められ、実行完了までに要する時間の長さには厳しい制約が課せられる。

【0016】上述の従来技術は、合成するプログラムの処理結果が、目標とする処理結果にどれだけ一致するか、という意味での性能向上を提供する技術であって、上述したような、課題の性格に沿った意味で性能が高いプログラムを合成することはできなかった。

【0017】

【発明が解決しようとする課題】本発明は上述した事情に対処すべくなされたものである。

【0018】処理すべき課題が与えられた際、プログラムの大まかな処理手順（概略処理手順）は容易に指定できる場合が多い。概略処理手順を、プログラム部品の組合せの制約条件として利用することで、試行錯誤の回数を減らし、より短時間に効率的に、性能の良いプログラムを合成することができるであろう。

【0019】この点に着目し、本発明は、概略処理手順を使って、プログラム部品の選択肢を一部分に限定することにより、効率よく（短時間に容易に）所望のプログラムを合成できるプログラム合成方法およびそれをを用いたプログラム合成装置を提供することを第1の目的とする。

【0020】また、本発明は、評価尺度を定義する手段を備えることにより、課題の性格に沿った意味で性能が高いプログラムを合成できるプログラム合成方法およびそれをを用いたプログラムの合成装置を提供することを第2の目的とする。

【0021】また、本発明は、高性能のプログラムを容易に合成できるプログラム合成方法およびそれをを用いたプログラムの合成装置を提供することを第3の目的とする。具体的には、

- ・ 合成したプログラムをターゲットマシン上で実行するのにかかる時間を評価基準としてプログラムの評価値を決定することで、より所望の時間に近い処理時間のプログラムを合成できるプログラム合成方法およびプログラム合成装置を提供する。

- ・ 【0022】合成したプログラムをターゲットマシン上で実行するときのプログラムコードのサイズを評価基準としてプログラムの評価値を決定することで、より小さいプログラムを合成できるプログラム合成方法およびプログラム合成装置を提供する。

- ・ 【0023】合成したプログラムをターゲットマシン上で実行するとき要する記憶領域のサイズを評価基準としてプログラムの評価値を決定することで、より小さいプログラムを合成できるプログラム合成方法およびプログラム合成装置を提供する。

- ・ 【0024】使用するプログラム部品の主観的優先度（例えば、部品の信頼性、拡張性のよさ、主観的評価等）を評価基準としてプログラムの評価値を決定することで、優先度の高い部品を多く使ってプログラムを合成できるプログラム合成方法およびプログラム合成装置を提供する。

【0025】

【課題を解決するための手段】本発明の第1のプログラム合成装置は、プログラム部品を組み合わせて所望のプログラムを合成するプログラム合成装置において、予め分類されたプログラム部品を記憶する部品記憶手段と、前記プログラム部品の分類に基づき所望のプログラムの

概略処理手順を指定する処理手順指定手段と、前記部品記憶手段に記憶されたプログラム部品から前記概略処理手順に対応するプログラム部品を選択する部品選択手段と、この部品選択手段で選択されたプログラム部品を結合してプログラムを合成する合成手段と、この合成手段で合成されたプログラムの性能を評価する評価手段と、を具備し、前記部品選択手段は、前記評価手段で得られた評価値を参照してプログラム部品を選択することにより、効率よく（短時間に容易に）所望のプログラムを合成できる。

【0026】また、本発明の第2のプログラム合成装置は、プログラム部品を組み合わせて所望のプログラムを合成するプログラム合成装置において、予め分類されたプログラム部品を記憶する部品記憶手段と、予め指定された条件に従って、前記プログラム部品の分類に基づき所望のプログラムの概略処理手順を生成する処理手順生成手段と、前記部品記憶手段に記憶されたプログラム部品から前記概略処理手順に対応するプログラム部品を選択する部品選択手段と、この部品選択手段で選択されたプログラム部品を結合してプログラムを合成する合成手段と、この合成手段で合成されたプログラムの性能を評価する評価手段と、を具備し、前記処理手順生成手段は前記評価手段で得られた評価値を参照して概略処理手順を生成し、前記部品選択手段は前記評価値を参照してプログラム部品を選択することにより、効率よく（短時間に容易に）所望のプログラムを合成できる。

【0027】また、本発明の第3のプログラム合成装置は、前記評価手段でプログラムを評価する際的评价尺度を定義する手段をさらに具備することにより、所望するプログラムの課題に沿った、性能の高いプログラムを合成できる。

【0028】また、本発明の第4のプログラム合成装置は、前記プログラム部品の分類に基づき所望のプログラムの概略処理手順の全部あるいは一部を指定する処理手順指定手段をさらに具備することにより、所望するプログラムの課題に沿った、性能の高いプログラムを合成できる。

【0029】また、本発明の第5のプログラム合成装置は、前記部品選択手段は、遺伝的アルゴリズムを用いて、前記部品記憶手段に記憶されたプログラム部品から前記概略処理手順に対応するプログラム部品を選択することにより、効率よく（短時間に容易に）所望のプログラムを合成できる。

【0030】また、本発明のプログラム合成装置は、評価手段で、合成手段で合成されたプログラムの性能、すなわち、プログラムの実行時間、プログラムサイズ、プログラムを実行する際に要する記憶容量をそれぞれ評価することにより、ユーザの所望する条件を満たした高性能のプログラムを容易に合成できる。

【0031】また、本発明の第1のプログラム合成方法

は、プログラム部品を組み合わせることで所望のプログラムを合成するプログラム合成方法において、前記プログラム部品の分類に基づき所望のプログラムの概略処理手順が指定されると、前記概略処理手順に対応するプログラム部品を選択し、この選択されたプログラム部品を結合してプログラムを合成し、この合成されたプログラムの性能の評価を行い、その評価値を前記概略処理手順に対応するプログラム部品を選択する際に参照することにより、効率よく（短時間に容易に）所望のプログラムを合成できる。

【0032】また、本発明の第2のプログラム合成方法は、プログラム部品を組み合わせることで所望のプログラムを合成するプログラム合成方法において、予め指定された条件に従って、前記プログラム部品の分類に基づき所望のプログラムの概略処理手順を生成し、この生成された概略処理手順に対応するプログラム部品を選択し、この選択されたプログラム部品を結合してプログラムを合成し、この合成されたプログラムの性能の評価を行い、その評価値を前記概略処理手順を生成する際、および前記概略処理手順に対応するプログラム部品を選択する際に参照することにより、効率よく（短時間に容易に）所望のプログラムを合成できる。

【0033】また、本発明の第3のプログラム合成方法は、前記合成されたプログラムの性能を評価する際の評価尺度が定義されることにより、所望するプログラムの課題に沿った、性能の高いプログラムを合成できる。

【0034】また、本発明の第4のプログラム合成方法は、前記プログラム部品の分類に基づき所望のプログラムの概略処理手順の全部あるいは一部を指定することにより、所望するプログラムの課題に沿った、性能の高いプログラムを合成できる。

【0035】また、本発明の第5のプログラム合成方法は、遺伝的アルゴリズムを用いて、前記概略処理手順に対応するプログラム部品を選択することにより、効率よく（短時間に容易に）所望のプログラムを合成できる。

【0036】また、本発明のプログラム合成方法は、合成されたプログラムの性能、すなわち、プログラムの実行時間、プログラムサイズ、プログラムを実行する際に要する記憶容量をそれぞれ評価することにより、ユーザの所望する条件を満たした高性能のプログラムを容易に合成できる。

#### 【0037】

【発明の実施の形態】以下、図面を参照して本発明の実施形態を説明する。

【0038】（第1の実施形態）図1は、本発明の第1の実施形態に係るプログラム合成方法を用いたプログラム合成装置の構成例を概略的に示したものである。本発明のプログラム合成方法は、ソフトウェアによって構成し、それを例えばパーソナルコンピュータに実装し、プログラム合成装置として実現してもよい。

【0039】図1において、プログラム合成装置は、処理手順指定手段1、部品選択手段2、プログラム部品記憶手段3、プログラム結合手段4、プログラム評価手段5から構成される。

【0040】図2は、プログラム合成装置の処理の流れを大まかに説明する流れ図である。

【0041】まず、処理手順指定手段1において概略処理手順が指定される（ステップS101-1）。プログラムの合成は、定められた終了条件が成立するまで繰り返される。（合成ループ；ステップS101-2～S101-6）。終了条件の例には、繰返し回数が予め与えられた上限値を越えたときに終了する条件や、合成するプログラムの評価値が予め与えられた目標値を越えたときに終了する条件、合成したプログラムの個数が予め与えられた上限個数を越えたときに終了する条件、あるいはこれらの条件の任意の組合せの論理和など様々に考えられる。ただし本発明の効果は、この終了条件の如何に拘らず発生し、その内容に拘束されるものではないことを指摘しておく。プログラム合成の繰返しごとに、ステップS101-3～ステップS101-5が実行される。まず、部品選択手段2は、指定された概略処理手順に従って、プログラム部品記憶手段3からプログラム部品を選択する（ステップS101-3）。つぎにプログラム結合手段4が選択されたプログラム部品を結合し、1本のプログラムに組み上げる。（ステップS101-4）。さらにプログラム評価手段5が、組み上がったプログラムの評価値を決定する（ステップS101-5）。次に各手段ごとに詳しく説明して行く。

#### 【0042】（1-1） 処理手順指定手段

図1において、処理手順指定手段1は、所望のプログラムがとるべき概略処理手順を指定するためのものである。処理すべき課題が与えられた際、プログラムがとるべき概略処理手順は、容易に指定できる場合が多い。概略処理手順を使って、プログラム部品の選択肢を一部分に限定することにより、試行錯誤の回数を減らし、より短時間に効率的に、性能の良いプログラムを合成することができる。

【0043】処理手順指定手段1の具体例として、あらかじめ保持している概略処理のリストを熟練者に提示し概略処理を選択させることによって、概略処理の手順を作成する例を説明する。図3に処理手順指定手段1の構成例を示す。

【0044】処理手順指定画面の表示手段1aは、概略処理のリスト（一覧表）が保持されている概略処理記憶手段1bから処理の一覧表を読み出して、ディスプレイ100上に表示する。熟練者は、作成しようとするプログラムの概略処理の手順を一覧表から選択して所望の概略処理の一連の手続きを作成する。

【0045】図4は、概略処理記憶手段1bに記憶されている概略処理の一覧表の記憶例を示している。図4に



において、各概略処理にはそれぞれを識別するための識別子が与えられている。なお、図4では、処理の名称（「概略処理1」、「概略処理2」、…）が識別子になっている。

【0046】次に、処理手順指定手段1に具備される、熟練者が概略処理手順を作成するためのユーザインタフェースの機能について説明する。

【0047】図5は、このインタフェースの例として処理手順指定手段1の例えば、処理手順指定画面の表示手段1aの制御のもと、ディスプレイ100に表示されるインタフェース画面の一例を示したものである。

【0048】図5において、概略処理手順を指定するためのインタフェース画面は、主に、概略処理記憶手段1bに記憶された概略処理の一覧を表示するウインドウ200と、このウインドウ200に表示された一覧から所望の概略処理を選択して概略処理の手順を編集するためのウインドウ201が表示される。

【0049】熟練者は、例えば、マウスなどの入力装置を使って概略処理を表す箱型のシンボル202をウインドウ201上の作業画面に張り付け、各々に対して、ウインドウ200に表示された概略処理の一覧からマウス等を用いて所望の概略処理を選択して割り付ける。

【0050】概略処理の手順は、概略処理を表す箱型のシンボル202を、実行の順序に従って矢印で結ぶことによって指定する。手順を作成するこの作業は、いわゆるフローチャートを作成する作業と同じである。また図中の六角形のシンボル203は、プログラミングの分岐処理をあらわすシンボルである。

【0051】（1-2） プログラム部品記憶手段  
プログラム部品記憶手段3は、プログラム部品を部品分類に分けて保持するものである。プログラム部品は、汎用に使えよう標準化したプログラムコードと、プログラムコードの属性からなる。プログラムコードの属性には、そのプログラムコードの用途、効果、各種のターゲットマシン上での実行にかかる時間、サイズ、引数の情報、プログラム部品の信頼性等を考慮した好ましさを表す使用優先度情報など、そのプログラムコードに関連する任意の情報を含めることができる。一つの完全な（実行可能な）プログラムは、複数のプログラム部品が結合されて成る。

【0052】「プログラム部品を結合する」とは、ここでは、プログラムコードの実行順序を決め、一方のプログラム部品の出力引数と、他方のプログラム部品の入力引数を接続し、どの引数からどの引数へと値が受渡しされるかを定めることである。

【0053】なお、部品分類とは、特定の観点でプログラム部品を部分集合にまとめる際の観点、または、分類そのものを指す。その一例としては類似の機能を持つプログラム同士をひとまとめにする部品分類が挙げられる。

【0054】また、概略処理とは、具体的には、プログラム部品を、例えば、その機能あるいはその効果に基づき分類し、その分類されたプログラム部品全体に共通する処理名である。ここでは、プログラム部品を分類した際の部品分類のタイトルが概略処理のタイトルである場合を例にとり説明している。

【0055】図6は、プログラム部品記憶手段3における、プログラム部品の記憶例を示したものである。図6において、プログラム部品は、例えば、部品分類10、331、510などに分類されて保持されている。

【0056】また、各プログラム部品は、図7に示すように、プログラムコードとその属性情報から構成されて記憶されるようになっている。

【0057】（1-3） 部品選択手段

部品選択手段2は、処理手順指定手段1が与える概略処理手順に含まれる個々の概略処理を、プログラム部品記憶手段3から選択したプログラム部品で置き換えるようになっている。

【0058】部品選択手段2の具体例として、処理手順指定手段1の概略処理記憶手段1bに記憶される概略処理の一覧と、プログラム部品記憶手段3の部品分類とが一对一に対応する場合を説明する。この例では、部品分類が類似の機能を持つプログラム同士をひとまとめにする分類であり、処理手順指定手段1に保持されている概略処理は、プログラム部品の機能的な分類に一致する。

【0059】部品選択手段2は、処理手順指定手段1から与えられる概略処理手順に含まれる個々の概略処理の並びに従って、対応する部品分類のプログラム部品を配列し、具体的な処理の手順を生成する。

【0060】図8を参照して、部品選択手段2の動作を説明する。

【0061】図8において、例えば、図5のウインドウ201上で編集された概略処理手順の個々の処理と、プログラム部品記憶手段3に格納されている部品分類との対応は、一点鎖線で表されている。

【0062】上述したように、部品選択手段2は個々の概略処理の並びに従って、対応する部品分類の中のプログラム部品を配列し、具体的な処理の手順を生成する。

【0063】本発明は部品選択手段2におけるプログラム部品の選択方法を特に限定するものではない。例えば、乱数を用いてプログラム部品を選択する方法でも良いし、また単に乱数で選択するのではなく、過去に合成したプログラムの評価値をプログラム評価手段5から取り込み、評価値の高低とプログラムに使われているプログラム部品との相関から、できるだけ高い評価値が得られるように、プログラム部品を選択する方法でも良い。また、利用者が置き換えを指示しても良い。この方法を用いると、試行錯誤的に、徐々に評価値の高いプログラムを合成することができる。

【0064】図8に示すように、プログラム選択手段2

でのプログラム部品の選択の結果、太枠で囲まれたプログラム部品が選択されたとする。

【0065】図9に、部品選択手段2による概略処理の置き換えの結果を示す。

【0066】(1-4) プログラム結合手段

プログラム結合手段4は、部品選択手段2が与える具体的な処理手順に従ってプログラムコードの実行順序を決め、一方のプログラム部品の出力引数と、他方のプログラム部品の入力引数を接続し、どの引数からどの引数へと値が受渡しされるかを定める。

【0067】プログラム結合手段4の具体例を説明する。本発明の効果は引数対応の具体的な方法に拘らず、引数対応の方法を特に限定する必要はない。以下に示す方法は一例である。プログラム部品記憶手段3の説明で述べたように、プログラム部品はプログラムコードとその属性から成る。プログラムコードの属性には、引数に関する情報が含まれている。引数対応は、引数に関する情報のうち引数の入出力種別と引数の内容を参照して決定する。

【0068】図10は、図9の4つのプログラム部品の引数の情報を示している。内容の項目には、その引数の内容を表す記号が入っている。

【0069】図11に示すフローチャートは、プログラム結合手段4が行なうデータ受渡しの指定方法の処理手順を示したものである。図10に示した4つのプログラム部品からなる具体的な処理手順に対してこのフローチャートに示す手続きを施すと、図12に示すようにデータの受渡しが決まる。矢印は、その根本でデータの送り手を示し、矢尻でデータの受け手を示している。

【0070】図11に示すフローチャートの手続きを簡単に説明すると、まず、あるプログラム部品iの引数jと、そのプログラム部品iより前にあるプログラム部品kの全引数lの内容が一致するかどうかをチェックして、一致する場合は、引数jを引数lと対応させる(ステップS3～ステップS8)。以上の処理をプログラム部品iに含まれる全ての引数について行い(ステップS2)、さらに、ステップS2～ステップS8の処理を全てのプログラム部品について行う(ステップS1)。

【0071】たとえば、プログラム部品137bの引数ipは、プログラム部品510-cの引数jrtの値を受け取る。入力引数あるいは入出力引数(入出力種別が、inputかin/out)であって、値を他のどの引数から得るかが定まらない引数のあるプログラム部品が残ってしまった場合には、その部品が置き換わった概略処理を、同じ概略処理に属する別のプログラム部品で置き換えて、上記の手順を再度やり直す。このようなすべての引数のデータの受渡し指定が決まると合成すべきプログラムが完成する。

【0072】(1-5) プログラム評価手段

プログラム評価手段5は、プログラム結合手段4でプロ

グラム部品が結合されて合成されたプログラムの評価値を決定するようになっている。

【0073】プログラム評価手段5の具体例として、プログラムを実行したときの実際の出力と、プログラムに期待されている出力との一致の度合に基づいて、プログラムの評価値を決定するプログラム評価手段を説明する。

【0074】図13に、プログラム評価手段5の構成例を示す。図13において、プログラム評価手段5は、プログラム実行手段5a、評価値計算手段5b、入力保持手段5c、期待出力保持手段5dから構成されている。

【0075】プログラム実行手段5aは、部品結合手段2から与えられる具体的な処理手順に従って、プログラム部品記憶手段3にあるプログラム部品を順次実行する。その際、プログラム部品間の変数の受け渡しは、プログラム結合手段4から与えられるプログラム部品間の引数対応に従って行われる。この時、入力保持手段5cが保持する入力データがプログラム実行手段5aで実行中のプログラムに入力される。その入力データを処理した結果である出力データは評価値計算手段5bに与えられる。

【0076】評価値計算手段5bは、入力データがプログラムに入力されたときにその処理結果として出力されることが期待されている出力データを、期待出力保持手段5dから読み出す。次に、出力データと期待されている出力データとを比較し、一致しているかどうかを調べ、それをもとに評価値を計算して出力する。

【0077】本発明は、評価値の計算方法には依存せず、その効果を発生するので、具体的な計算式に拘束されるものではない。

【0078】ここで、評価値の計算方法の一例を説明する。図14は、入力データとそれがプログラムに入力されたときに期待される出力データと、合成したプログラムの実際の出力データとの対応の一例を示したものである。この例では、入力データは全部でn件ある。出力データO<sub>i</sub>と期待される出力データO<sub>exp</sub>(i)が一致する件数をmとしたとき、プログラムの評価値eを $e = m/n$ で求める。すなわち、入力データ件数に対する、出力データと期待される出力データの一致件数の比率を評価値とする。

【0079】計算された評価値は部品選択手段2に送られ、次のプログラム合成において、より評価値の高いプログラムを合成するには概略処理の各々にどのプログラム部品を選択すれば良いかを判断するために使われる。

【0080】(1-2)で説明したプログラム部品記憶手段は、プログラム部品を部品分類に分けて保持していたが、プログラム部品を部品分類と対応づけて保持してもよい。

【0081】すなわち、プログラム部品はプログラムコードとその属性から成るが、図15に示すプログラム部

品の記憶例のように、属性情報の中に、部品分類の情報が含まれていてもよい。

【0082】以上説明したように、本実施形態によれば、プログラム部品の組合せの制約条件としてユーザが概略処理手順を指定し、その指定手順に従ってプログラム部品を結合し、これを評価するものとなっているため、試行錯誤の回数が減少することになり、より短時間に効率的に、性能の良いプログラムを合成することができる。

【0083】（第2の実施形態）次に、本発明の第2の実施形態として、プログラムを評価する尺度（方法）を定義する評価尺度定義手段をさらに具備したプログラム合成装置について説明する。

【0084】図16に示すように、第2の実施形態に係るプログラム合成装置の構成例は、図1の第1の実施形態に係るプログラム合成装置にさらに評価尺度定義手段6が追加されて構成される。

【0085】評価尺度定義手段6は、プログラム評価手段5でプログラムの評価値を計算する際の計算方法を与えるものである。

【0086】評価尺度定義手段6の具体例として、合成したプログラムを評価するための評価プログラムを供給するものを説明する。図17は、この評価尺度定義手段6とプログラム評価手段5の関係を示した図である。この具体例では、合成したプログラムの評価は、以下のデータを評価用入力変数とする関数によって計算する。

- ・合成したプログラムのソースコード
- ・合成したプログラムのソースコードのサイズ
- ・合成したプログラムが要する記憶領域のサイズ
- ・合成したプログラムに入力データを処理させたときの出力データ
- ・期待されている出力データ
- ・合成したプログラムに使われたプログラム部品の属性情報
- ・合成したプログラムの実行時間

評価尺度定義手段のソースコード編集手段6aは、上記の評価用入力変数を入力データとし、合成したプログラムの評価値を算出する任意の評価プログラムソースコードを編集するためのものである。このソースコード編集手段6aを人間が操作し、所望の評価尺度を実現する評価プログラムソースコードを作成する。作成されたソースコードはコンパイラ6bに渡され、実行可能な評価プログラムに変換される。この評価プログラムは、プログラム評価手段5の評価値計算手段5bに供給される。

【0088】同図に示すように評価値計算手段5bには、上述の評価用入力変数に相当するデータが入力されており、評価プログラムが評価値計算手段5bによって実行されることにより、合成したプログラムの評価値が算出される。算出された評価値は、部品選択手段2に供給される。

【0089】つぎに、人間がソースコード編集手段6aで作成する評価プログラムで計算される評価関数の例をいくつか示す。

【0090】（実際の出力データと期待されている出力データとの違いによる評価）第1の実施形態で説明した評価値 $e$ を、上述の評価尺度定義手段6によって提供される評価プログラムで計算できることは明らかである。それを以下に示す。入力保持手段5cに保持される複数件数のテスト入力データを順次、プログラム実行手段5aで実行されている合成したプログラムに入力すると、その件数分の実際の出力データが算出される。今、その件数を $n$ 件としよう。この件数は、合成したプログラムが評価値計算手段5bに供給する出力データの件数と一致しているので、入力保持手段5cから直接供給されなくても、評価値計算手段5bは知ることができる。また個々のテスト入力データに対応する期待されている出力データと、実際の出力データとが一致する件数 $m$ は、評価値計算手段5bに入力されている実際の出力データの個々と、これらに対応する期待されている出力データの個々とを比較することで求められる。したがって第1の実施形態で説明した評価値の例は、上述の評価尺度定義手段6が供給した評価プログラムによって算出できる。

【0091】（実際の出力データと期待されている出力データとの誤差による評価）つぎに数学関数のフィッティングに適した評価値として、テスト入力データに対する出力データと、期待される出力データとの誤差の例を示す。全部で $n$ 件あるうちの $i$ 番目の入力テストデータに対する実際の出力データを $O(i)$ 、期待されるデータ出力を $O_{exp}(i)$ とすると、評価関数 $e_2$ を次のように定める。

【0092】

【数1】

$$e_2 = \frac{1}{1 + 1/2 \sum_{i=1}^n (O(i) - O_{exp}(i))^2}$$

【0093】誤差が小さいほど最大値1.0に近い評価値が得られる。この評価値は、実際の出力データ、期待される出力データ、および入力データの件数のみで計算できるから、上述の評価尺度定義手段6が供給する評価プログラムとして実現することが可能である。

【0094】（計算結果が期待する値に一致し、プログラムの実行時のサイズがより小さいものを高く評価する場合）次の例として、合成したプログラムの出力が期待されている出力に完全に一致することを絶対の条件とし、さらに、よりプログラムの実行時のサイズが小さいものを高く評価する場合の例を示す。プログラム実行手段5aから評価値計算手段5bに供給される、合成したプログラムの実行時のサイズを $s$ としよう（単位はバイト）。このとき、求める評価値 $e_3$ を次のように定める。

【0095】

【数2】

$$e_3 = \begin{cases} 1/(1+s) & (e_2 = 1.0 \text{ のとき}) \\ 0 & (\text{その他}) \end{cases}$$

【0096】ここで $e_2$ は前述の評価関数とする。この評価値 $e_3$ は、実際の出力データ、期待される出力データ、入力データの件数、および合成したプログラムのサイズのみで計算できるから、上述の評価尺度定手段6が供給する評価プログラムとして実現することが可能である。

【0097】（計算結果が期待する値に一致し、実行に要する時間がより短いものを高く評価する場合）次の例として、合成したプログラムの出力が期待されている出力に完全に一致することを絶対の条件とし、さらに、より実行に要する時間が短いものを高く評価する場合の例を示す。評価値計算手段5bには、時計5eから時刻が供給されている。プログラム実行手段5a上で合成したプログラムが実行を開始した時刻と、実行を終了した時刻とを使って、所要時間 $t$ を求められる。したがって前述の評価値 $e_3$ の定義式において、プログラムの実行時のサイズ $s$ を $t$ に置き換えると、ここで求める新しい評価関数 $e_4$ が得られる。

【0098】（計算結果が期待する値に一致するテスト入力データの比率と、合成したプログラムの品質メトリクスとの線形和を評価値とする場合）次の例として、実際の出力と期待する出力とが一致するテスト入力データの件数の全テストデータに対する比率と、合成したプログラムのソースコードの品質メトリクスの線形和として評価値を求める場合を示す。実際の出力と期待する出力とが一致するテスト入力データの件数の、全テストデータ件数に対する比率は前述の評価値 $e$ である。合成したプログラムのソースコードの品質メトリクス $m$ はプログラムソースコードの品質を定量化した数値であり、様々なものが一般に知られている（たとえば、森口繁一編「ソフトウェア品質管理ガイドブック」日本規格協会刊、第12章品質の定量的計測と評価の方法を参照）。ここでは、ソースコードの複雑さを計測する尺度である条件文数の総数 $i$ の関数として $m = 1 / (i + 1)$ とする。このとき、出力一致度の百分率 $c$ を8割、品質メトリクス $m$ を2割として評価値 $e_5$ を定義すると、 $e_5 = 0.8c + 0.2m$ となる。評価関数 $e_5$ も、評価尺度定義手段6がプログラム評価手段5に供給する評価プログラムとして実現することができることは、図17から明らかであろう。

【0099】（その他）その他評価関数を定義する際に、入力変数としてプログラム部品の属性情報として各部品に予め付けられた主観的優先度（後述する第6の実施形態で詳しく述べる）を使ってもよい。

【0100】また、ここでは前述の六種類の評価用デー

タを入力変数とする関数として評価関数を定義する例を説明したが、本発明の評価尺度定義手段6は評価値を計算する関数の入力変数には依存しないので、何を入力変数とするかには拘束されずその効果を発揮することに注意されたい。

【0101】（第3の実施形態）前述の第1、第2の実施形態のプログラム合成装置では、設計者が処理手順指定手段1を用いて概略処理の手順を作成していく。ここでは、この概略処理の手順を自動的に生成する処理手順生成手段7を具備したプログラム合成装置について説明する。

【0102】図18に第3の実施形態に係るプログラム合成装置の構成例を示す。なお、図18において図1と同一部分には同一符号を付し、異なる部分について説明する。すなわち、図1の処理手順指定手段1が処理手順生成手段7に置き換わっている。

【0103】処理手順生成手段7は、図19に示すように、処理手順発生手段7aと概略処理記憶手段7bと処理手順長入力手段7cから構成される。

【0104】処理手順長は人間が指示する正の整数 $N$ であり、処理手順の最大の長さを意味している。

【0105】処理手順発生手段7aは、処理手順長入力手段7cで指定された長さ $N$ 以下の概略処理のシーケンス（手順）を概略処理記憶手段7bに記憶されている概略処理からランダムに発生し、部品選択手段2に出力する。以降、第1の実施形態の説明と同様にして、プログラム合成を行い、プログラム評価手段5でその評価値を算出し、その算出された評価値が、処理手順発生手段7aに入力されると、それを上記の概略処理のシーケンスの評価値とし、更に別の概略処理のシーケンスを発生させて同様に評価値を得ることを繰り返し、評価値の高い概略処理のシーケンスとプログラム合成されたプログラムを得る。

【0106】上記では、指定された処理手順長に基づき概略処理のシーケンスをランダムに発生する場合を示したが、各種の組み合わせ探索手法（整数計画法、遺伝的アルゴリズム、タブーサーチ等を用いることもできる）。

【0107】（第4の実施形態）次に、前述の処理手順指定手段1と処理手順生成手段7を併用するプログラム合成装置について説明する。

【0108】図20に第4の実施形態に係るプログラム合成装置の構成例を示す。なお、図1と異なる部分は、処理手順指定手段1にさらに処理手順生成手段7が追加されている。

【0109】ユーザは所望の概略処理手順のうちの一部を、前述同様に処理手順指定手段1で指定を行い、前記概略処理手順のうち、自動的に選択して欲しい処理手順は処理手順生成手段7が生成するものである。

【0110】処理手順指定手段1で指定した以外の部分

を前述の第3の実施形態と同様に処理手順生成手段7で発生し、以降の処理は前述同様である。

【0111】また、プログラム評価手段5で得られる評価値は処理手順指定手段1にも返され、ユーザがそれを参考に概略処理手順の再指定を行い、修正する。

【0112】（第5の実施形態）第1の実施形態では、プログラム部品記憶手段3に保持されている概略処理と部品分類とが図8に示すように、一対一に対応している場合について説明した。プログラム部品の管理の便宜で作成する部品分類の場合、必ずしも部品分類と概略処理とは一対一に対応付けすることができず、一般的には多対多の対応となる。

【0113】ここでは、処理一部品分類項目対応テーブル20を持ち、多対多の対応に対処できるプログラム合成装置について説明する。

【0114】プログラム合成装置の構成例としては、図1と同様であるが、例えば、部品選択手段2あるいはプログラム部品記憶手段3には、処理一部品分類項目対応テーブル20を具備しているものとする。

【0115】図21を参照して、処理一部品分類項目対応テーブル20を用いた場合の動作を説明する。すなわち、部品選択手段2は、例えばウィンドウ201上で指定された概略処理手順の個々のステップを、処理一部品分類項目対応テーブル20で対応づけられる部品分類の中のプログラム部品と置き換えて、具体的な処理の手順を生成するようになっている。

【0116】（第6の実施形態）次に、プログラム評価手段5で合成されたプログラムを評価する際、その合成されたプログラムを実行するのにかかる時間、必要な記憶容量、プログラムコードのサイズ等を評価基準として、評価値を決定するプログラム合成装置について説明する。すなわち、プログラム評価手段5は、プログラム部品記憶手段3からプログラム部品に関する必要な情報を受け取り、評価値を算出する。

【0117】図22にプログラム部品記憶手段3に保持されるプログラム部品の属性情報の記憶例を示す。

【0118】図22に示すように、各プログラム部品毎に、ターゲットマシン上での実行時間、ターゲットマシン上でのプログラムサイズ、ターゲットマシン上での実行に要する記憶容量、および、各プログラム部品の主観的優先度（例えば、部品の信頼性、拡張性のよさ、主観的評価等）を考慮した利用の好ましさを表す優先度の情報が記憶されている。

【0119】プログラム評価手段5は、プログラム結合手段4から渡された合成プログラムを構成している各プログラム部品の、ターゲットマシンに対応する情報や優先度を、プログラム部品記憶手段3の図22に示したようなテーブルから読み出し、例えば、その合計を算出することにより、ターゲットマシン上での全実行時間、全プログラムサイズ、全記憶容量、総合優先度を評価値と

して算出する。

【0120】なお、算出方法は種々の演算方式が考えられ、更に、評価値は単独で用いられるだけでなく、複数の評価値を組み合わせ、例えば加重和をとって、評価値とすることも考えられる。

【0121】（第7の実施形態）次に、処理手順指定手段1で指定された概略処理手順に基づきプログラム部品を選択する際に、遺伝的アルゴリズムを用いるプログラム合成装置について説明する。

【0122】図23は、第7の実施形態に係るプログラム合成装置の構成例を示したものである。図1と異なる部分は、部品選択手段2が遺伝的アルゴリズムを用いて部品探索を行う部品探索手段8に置き換わっている。

【0123】尚、遺伝的アルゴリズムについては、「遺伝的アルゴリズムハンドブック」（L. デービス編、森北出版）等に述べられているので、ここでは適用の仕方を述べる。

【0124】部品探索手段8では、処理手順指定手段1より与えられる概略処理手順に含まれる個々の概略処理の並びに従って、プログラム部品記憶手段3からプログラム部品を選択して配列する。この処理は第1の実施形態の説明と同様である。この配列処理によって一つの染色体（実行順序を持った一連のプログラム部品の流れを表す情報表現）が構成される。異なった配列方法で作られる複数の染色体とともに、染色体集団を構成する。この時、集団に属する染色体の個数は予め定められた整数とする。

【0125】染色体集団はプログラム結合手段4に渡され、プログラム結合手段4が個々の染色体について、第1の実施形態の説明と同様に、プログラム部品間の引数対応の決定を行ってプログラムに変換する。このプログラムはプログラム評価手段5に渡され、実行され、第1の実施形態の説明と同様にプログラムの評価値（染色体の評価値）が算出される。すべての染色体の評価値を求め、部品探索手段8にそれらすべての評価値を返す。

【0126】部品探索手段8は、より良い評価値を得る可能性の高い染色体を交叉や突然変異を用いて生成し直す。一つの行い方は、評価値に比例した確率で任意の2つの染色体を選び、これを両親として子供の染色体を交叉で産生する。例えば、概略処理手順に含まれる全ての個別の概略処理からなる集合を確率的に2つの部分A、Bに分け、父親のAの部分に含まれる概略処理に対応していたプログラム部品と母親のそれとを入れ替えることで交叉が実現され、子の染色体が産生される。その染色体には更に突然変異によって多様性を増加させる方法もある。例えば、確率的に1つの概略処理を選び、それに対応するプログラム部品を他のプログラム部品に確率的に置き直すことで実現される。

【0127】このように、新しい世代の染色体集団を作成し、同様に、プログラム結合手段4、プログラム評価

手段5を通し、個々の染色体の評価値を得る。このことを繰り返すことにより、染色体集団の中に、より評価値の高い染色体が構成されていく。

【0128】遺伝アルゴリズムにも種々の手法があり、それらを利用した方法も変形例として考えられる。また、探索手法としてタブーサーチ等の任意の組合わせ最適化手段も利用できる。

【0129】(第8の実施形態)次に、第7の実施形態に係るプログラム合成装置を、例えば、画像前処理プログラムの合成に適用する場合について説明する。題材の画像前処理プログラムは、工業製品の生産ラインで使われる自動目視検査装置に利用する。

【0130】図24は、画像前処理プログラムを合成するプログラム合成装置の構成例を示したものである。以下に各部の機能、動作を説明する。

【0131】(8-1) 処理手順指定手段

処理手順指定手段1は、概略処理手順入力部1cとそれにつながる端末1dから構成される。概略処理手順入力部1cには、図4に示した概略処理の一覧表が記憶されている。概略処理の一覧表の具体例を図25に示す。すなわち、概略処理の一覧表として、画像前処理プログラムを合成するのに必要な概略処理の名称の一覧が記憶されている。

【0132】目視検査装置で用いる、例えば、画像前処理プログラムを開発する専門家は、サンプル画像を見ることによって、どんな処理を行えば目的の不良が検出できるか、容易に概略手順を決定できる。たとえば、図26に示す検査対象の表面上の色斑を検出したいとする。この不良は、色の急激な変化として検出できる。

【0133】この場合、概略処理手順は、一例として以下ようになる。

【0134】手順1) RGB系で表現されたカラー画像を、色情報変換によって別の色表現系(例えば、マンセル色空間など)に変換する。

【0135】手順2) 色が急激に変化している部分(色エッジ)を強調する。

【0136】手順3) 後の計算量を減らすため、三次元の値を一次元の値におとす。

【0137】手順4) 色斑の存在を、色エッジがあるか否かによって検出するため、連続した線の存在度を計算する。

【0138】最終的に、算出された連続した線の存在度が規定値以上の場合には、色斑があると判断する。

【0139】専門家は、概略処理手順入力部1cが端末1dに具備されるディスプレイ上に表示するインターフェースを使って概略処理手順を入力する。

【0140】図27は、インターフェース画面上に入力した上記の概略処理手順を示し、図5に対応するものである。

【0141】(8-2) プログラム部品記憶手段

プログラム部品記憶手段3は、プログラム部品と部品番号を保管する部品ファイルとして実現する。

【0142】ここでは、部品分類としては、機能的な類似性の観点による分類を採用する。部品分類、および各分類に属するプログラム部品の記憶例を図28に示す。この部品分類は、処理手順指定手段1に具備される概略処理の一覧表(図25参照)の各項目と、一対一に対応する例である(再度強調するが、部品分類と概略処理とは、必ずしも一対一に静的に対応している必要はない)。プログラム部品には、各分類中で一意の部品番号が付されている。

【0143】各プログラム部品は、プログラムコードとその属性情報との組で記憶されるようになっている。

【0144】図29は、「Prewitt空間フィルタ」、「色相・再度変換」というプログラム部品を例に、各プログラム部品の格納方法を示したものである。図29に示すように、各プログラム部品に対応して、ソースコードで表現されているプログラムコードと、プログラム属性の情報が格納されている。

【0145】「Prewitt空間フィルタ」のプログラム属性を見ると、このプログラムが、入力用引数3個と出力用引数1個を取ること、あるターゲットマシン上での標準サイズ画像の処理時間が4.7秒であること、プログラムのサイズが2813バイトであること、などが分かる。

【0146】この例でプログラムコードはソースコードとして表現されているが、表現の形式は問わないので例えばソースコードをコンパイルした結果であるオブジェクトコードであってもかまわない。

【0147】(8-3) 部品探索手段、プログラム結合手段、プログラム評価手段

本実施形態の部品探索手段はGA(遺伝アルゴリズム)技術を利用するもので、コーディング情報決定部8a、初期染色体発生部8b、交叉・突然変異操作部8c、部品決定部8d、淘汰操作部8eから構成される。

【0148】部品探索手段8は、プログラム結合手段4、プログラム評価手段5と密接に関係している。まず、上記3つの手段の動作を簡単に説明する。

【0149】ステップT1) 部品探索手段8は、プログラム部品を生物の染色体に各々見立てて、ビット列などのコードで表す。ついで乱数などで、初期の染色体集団を生成する。

【0150】ステップT2) 生成した染色体集団に対して、交叉と突然変異を施し、新しい染色体集団をつくる。これら新しい染色体集団は、プログラムを構成する部品のコードに変換され、プログラム結合手段4に渡される。

【0151】ステップT3) プログラム結合手段4は、プログラム部品のコードを結合しプログラム群を生成する。

【0152】ステップT4) 生成されたプログラム群は、プログラム評価手段5で実行され、各プログラムの評価値が計算される。

【0153】ステップT5) プログラムの評価値は部品探索手段8に供給される。ここで、評価値が高いプログラムを表現する染色体を残し、評価値の低い染色体を淘汰することによって、全体として進化した(評価値が高い)染色体集団を生成する。

【0154】ステップT6) 終了条件が成立したなら、プログラムの生成を終了する。成立していなければ、ステップT2に戻る。終了条件には、ステップT2～ステップT6の手続きを何世代繰り返したか、プログラムの最高評価値が規定の値に達したか、などを設定する。

【0155】このように、染色体の生成、染色体が表現するプログラムの生成、プログラムの評価、評価値による染色体淘汰のステップを繰り返すことで、染色体の評価値が徐々に向上し、したがって評価値の高い画像前処理プログラムが得られる。以下各ステップの詳細を説明する。

【0156】ステップT1) 部品探索手段8は、概略処理手順と部品ファイルを読み込む。

【0157】部品探索手段8のコーディング情報決定部8aは、概略処理手順と、部品ファイルに保管されている各概略処理に入るプログラム部品の種類の数とから、コーディング情報を生成する。図30は、コーディング情報が表現している各概略処理手順と、染色体ビット列との対応関係の例を示している。図27のウィンドウ201に表示されている概略処理手順と、図28に示したような部品ファイルに収められているプログラム部品の情報から生成されるコーディング情報の具体例を図30に示す。

【0158】図27に示す概略処理手順では、概略処理の数が4つで、第1の概略処理は「色情報変換」である。図28によると、「色情報変換」に対応するプログラム部品は5種類なので3ビットで区別できる。同様に残りの概略処理のプログラム部品は、「色エッジ強調」が4ビット、「一次元化」が3ビット、「直線の存在度計算」が9ビットでそれぞれ区別できる。したがって図30に示す通り、プログラムを表現する染色体は、全体

で19ビットの長さになる。

【0159】ここまでの処理によって、プログラムと染色体の対応、すなわち染色体のコーディングが決定したことになる。たとえば染色体

0101011110000000011

には、最初の3ビットで「色情報変換」が「YIQ変換」であること、つづく4ビットで「色エッジ強調」が「S-G空間フィルタ」であること、つづく3ビットで「一次元化」が「 $x_3$ 」であること、つづく9ビットで「直線存在度」が「マスク処理なしハフ変換分割数20」であることが表現されている。

【0160】次に、コーディング情報8fは初期染色体発生部8bに渡される。ここでは、予め決められるか、ユーザーが指定する数の染色体が乱数によって生成される。生成された染色体にプログラムが存在しない場合には、改めて生成し直す。たとえば生成された染色体が、111...

であるとする、最初の概略処理である「色情報変換」として部品番号「7」のプログラム部品を選んだことになるが、図30のコーディング情報によれば最初の概略処理の部品数は「5」であるから、この染色体が表現するプログラムが存在しないことになる。この場合は、染色体を生成し直す。

【0161】生成された染色体集団8gは交叉・突然変異操作部8cに供給され、ここで、交叉や突然変異の操作が施される。

【0162】交叉：染色体の間で切断と再結合が生じて部分交換が行われ、その結果、染色体組換えがおこることを交叉という。この染色体の操作は、優れた染色体の掛け合わせからより優れた染色体が生まれる場合がある、とする考え方を基本にしている。

【0163】いま、図32に示すように、

染色体A: 00001100100000000000と、

染色体B: 10000001100000000010を交叉させる場合を考える。図32を見ると分かるように、染色体Aは次のようなプログラムを表現している。

【0164】

【数3】

(色情報変換) 色相・彩度変換



(色エッジ強調) Laplacian空間フィルタタイプ1



(一次元化)  $\sqrt{x_2^2 + x_3^2}$



(直線存在度計算) マスクなしハフ変換、分割数12

【0165】染色体Bは次のようなプログラムを表現している。

【0166】  
【数4】

(色情報変換) 無変換



(色エッジ強調) Prewitt空間フィルタ



(一次元化)  $x_3$



(直線存在度計算)マスクなしハフ変換、分割数20

【0167】まず、染色体A、Bを複写しA'、B'を作る。ここでの交叉法は、二箇所切断し交叉する二点交叉法を用いることにする。切断点はランダムに選ぶが、仮に2箇所の切断点を図32に示すように、5ビット目と6ビット目の間、および、9ビット目と10ビット目の間に選んだとする。

染色体A'、B'は、図33に示すように変化する。

【0169】このように交叉によって、染色体Aと染色体Bとから、新しい染色体A'、B'が生成された。図33から明らかなように、染色体A'は次のようなプログラムを表現している。

【0170】

【数5】

【0168】次に、切断点の間の部分を交換すると、染

(色情報変換) 色相・彩度変換



(色エッジ強調) Sobelフィルタ3×3



(一次元化)  $\sqrt{x_2^2 + x_3^2}$



(直線存在度計算)マスクなしハフ変換、分割数12

【0171】染色体B'は次のようなプログラムを表現している。

【0172】

【数6】

(色情報変換) 無変換



(色エッジ強調) Kirsch空間フィルタ5×5



(一次元化)  $x_3$



(直線存在度計算)マスクなしハフ変換、分割数20

【0173】この場合は概略処理であるエッジ強調の部品が変更されただけであったが、切断点によっては、もっと大幅に部品が変化し得る。

【0174】ある染色体に交叉を施す確率は、予め決めるか、ユーザによって指定される交叉率で与える。交叉の相手になる染色体や切断点は、ランダムに選ぶ。また交叉させた結果存在しないプログラムの染色体が生成された場合は、それを捨てる。交叉はやり直さない。

【0175】突然変異：染色体の一部を別の染色体で置換し、新しい染色体をつくる操作を突然変異という。

突然変異は、多くの染色体が似通ってしまい、染色体集団として多様性が失われたときに起こる進化の停滞を、強制的に打ち破る効果を持つ。

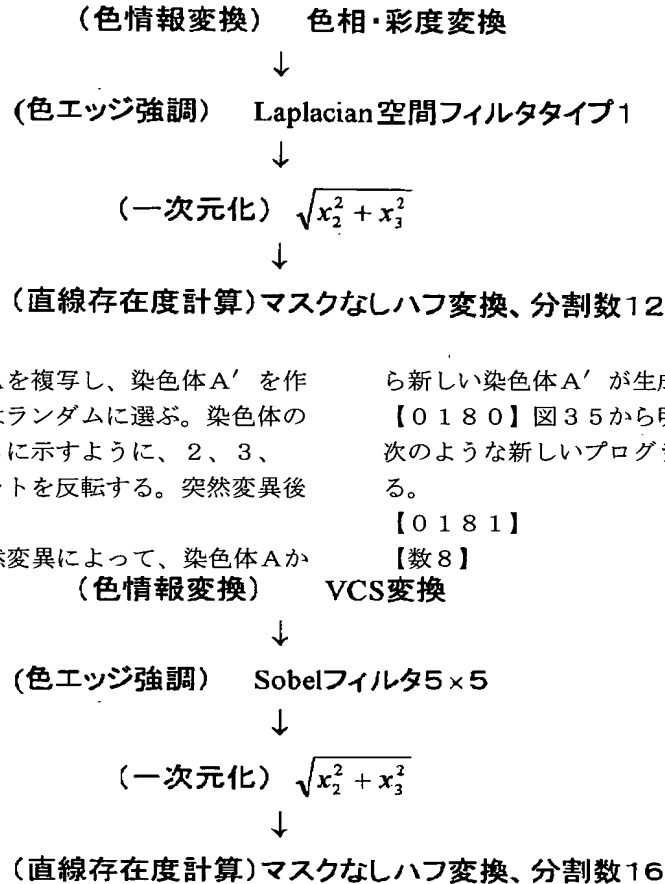
【0176】いま、図34に示すように、

染色体A： 00001100100000000000  
を突然変異させる場合を考える。染色体Aは、次のようなプログラムを表現している。

【0177】

【数7】





【0178】まず、染色体Aを複写し、染色体A'を作る。染色体を置換する位置はランダムに選ぶ。染色体の置換として、ここでは図34に示すように、2、3、6、7、19ビットめのビットを反転する。突然変異後の染色体を図35に示す。

【0179】このように突然変異によって、染色体Aか

ら新しい染色体A'が生成された。

【0180】図35から明らかなように、染色体A'は次のような新しいプログラムを表現していることが分かる。

【0181】

【数8】

【0182】この場合は、概略処理である色情報変換、色エッジ強調、直線存在度計算が変更された。

【0183】ある染色体の特定のビットを反転させる確率は、予め決めるかユーザによって指定される突然変異率で与える。また突然変異の結果存在しないプログラムの染色体が生成された場合は、それを捨てる。突然変異の操作はやり直さない。

【0184】このようにして交叉と突然変異で新しく生成した染色体を加えて、全体として数の増えた染色体集団(図24の染色体集団8h)を部品決定部8dに供給する。

【0185】部品決定部8dは、コーディング情報に従って染色体を解釈し、その染色体が表現しているプログラムのプログラム部品を部品ファイルから検索する。そして、染色体集団8hの各染色体ごと、プログラム部品、プログラム部品と概略処理との対応(どの概略処理をどのプログラム部品で置き換えるか)、概略処理の接続関係を組み合わせてすべての染色体についてまとめたものを、部品情報8iとしてプログラム結合手段4に供給する。

【0186】部品探索手段8は、以上のように、概略処理手順からプログラムの合成に必要な全ての情報を供給する。

【0187】ステップT3)、プログラム結合手段4は、

最終的には部品決定部8dが供給する具体的な処理手順(すなわち部品情報8i)にしたがって、プログラム部品のプログラムの引数を結合する。結合の方法については、第1の実施形態の説明と同様である。

【0188】プログラム結合手段4のプログラム合成部4a(図24参照)は、プログラム部品のプログラムの引数の結合に加え、必要な記憶領域の確保や、変数の宣言、画像データの読み込みに必要なソースコードの生成をおこなう。引数が結合されたプログラム部品と前述のソースコードとを接続すると、染色体が表現しているプログラムができあがる。

【0189】染色体集団8hの全ての染色体に対応するプログラムソースコード群4c(図24参照)は、コンパイラ・リンカ4bに渡されて、プログラム実行形式群4dに変換される。

【0190】ステップT4)生成されたプログラム実行形式群4dは、プログラム評価手段5のプログラム実行部5fに供給される。評価用データ5gには、画像データとその画像データに色斑があるか専門家が判定した結果との組が、複数(ここではn組とする)保持されている。

【0191】プログラム実行部5fは、評価用データの画像データをプログラムに投入し、そのデータに対するプログラムの出力値(この場合、画像に対する直線の存

在度)を受けとる。前述したように、画像処理プログラムによる色斑の有無の判断は、算出した直線の存在度が規定値以上であるか否かによって決める。このプログラムによる色斑の有無の判断と、専門家による判定結果とが一致する画像の件数を $m$ としたとき、このプログラムに対する評価値 $e$ を、

$$e = m / n$$

と算出する。

【0192】プログラム実行部5fは、プログラム実行形式群4dのすべてのプログラムについて評価値を算出する。

【0193】ステップT5) 算出されたプログラム評価値は、部品探索手段8の淘汰操作部8eに供給される。淘汰操作部8eは、染色体集団8hの個々の染色体に対応するプログラムの評価値の高い順に上位 $j$ 位までの染色体のみをのこし、それ以下の染色体を捨てる。ここで $j$ は、初期染色体発生日8bで最初に生成した染色体の数に等しい。残った染色体は、次の世代の染色体8gとして交叉・突然変異部8cに供給される。

【0194】ステップT6) ステップT6)については、前述の通りである。

#### 【0195】(8-4) 効果

プログラムを生成しては評価値を計算し、評価値の高いものを残して次の世代のプログラムをまた生成するという方式では、生成したプログラムを実際に実行しなければならない。前述したように、生成するプログラムが処理時間を要するものである場合(言うまでもなく、画像処理は時間が掛かる処理の典型である)、所望の評価値に達するプログラムを得るまでには、膨大な回数の繰り返し処理と、処理時間が必要となる。

【0196】処理時間を短縮するには、プログラム生成の際のプログラム部品の選択肢を、良いプログラムが生成できそうな範囲に制約することが非常に効果がある。しかしこれまでは、プログラム部品の選択肢への制約をかける方法が存在しなかった。

【0197】本発明は、プログラムを開発しようとしている専門家は、概略の処理手順なら比較的容易に設計できる場合が多いということに着目し、この概略処理をプログラム部品の制約に利用することによって、上記の問題を解決するものである。これまで、最初は暫定的なプログラムから初めて徐々に改良を加えることで性能の良いプログラムを合成する技術は、プログラム部品の組合せ爆発が原因で、現実的な時間内に所望性能のプログラムを生成できない、現実性の薄いものであった。これを初めて現実的なものにするという意味において、本発明の効果は非常に大きいといえる。

#### 【0198】

【発明の効果】以上説明したように、本発明によれば、効率よく(短時間に容易に)所望のプログラムを合成できる。また、所望するプログラムの課題に沿った、性能

の高いプログラムを合成できる。また、ユーザの所望する条件を満たした高性能のプログラムを容易に合成できる。

#### 【図面の簡単な説明】

【図1】本発明の第1の実施形態に係るプログラム合成装置の概略構成を示すブロック図。

【図2】プログラム合成装置の処理の流れを大まかに説明する流れ図。

【図3】処理手順指定手段の構成を示すブロック図。

【図4】図3の概略処理記憶手段に記憶されている概略処理の一覧表のデータ構造を示す図。

【図5】処理手順指定手段によりディスプレイに表示されるインタフェース画面の一例を示す模式図。

【図6】プログラム部品記憶手段のプログラム部品のデータ構造を示す図。

【図7】プログラム部品のデータ構造を示す図。

【図8】部品選択手段によるプログラム部品の選択動作を説明するための図。

【図9】部品選択手段により概略処理が適当なプログラム部品に置き換えられた結果の一例を示す図。

【図10】プログラム部品記憶手段で記憶されるプログラム部品の属性情報の一例を示す図。

【図11】プログラム結合手段が行なうデータ受渡しの指定方法の処理手順を示すフローチャート。

【図12】図11に示した手順に従って求められた各プログラム部品間のデータの受渡しの様子を示す図。

【図13】プログラム評価手段の概略構成を示すブロック図。

【図14】プログラム評価手段で合成されたプログラムを評価する際合成されたプログラムに入力された入力データと、それがプログラムに入力されたときに期待される出力データと、合成したプログラムの実際の出力データとの対応の一例を示す図。

【図15】プログラム部品記憶手段に記憶されているプログラム部品のデータ構造の他の例を示す図。

【図16】本発明の第2の実施形態に係るプログラム合成装置の概略構成を示すブロック図。

【図17】評価尺度定義手段とプログラム評価手段の関係を示した図。

【図18】本発明の第3の実施形態に係るプログラム合成装置の概略構成を示すブロック図。

【図19】処理手順生成手段の構成を示すブロック図。

【図20】本発明の第4の実施形態に係るプログラム合成装置の概略構成を示すブロック図。

【図21】本発明の第5の実施形態に係るプログラム合成装置の置部品選択手段による概略処理とプログラム部品との対応付けの様子を示す図。

【図22】本発明の第6の実施形態に係るプログラム合成装置のプログラム部品記憶手段に保持されるプログラム部品の属性情報の一例を示す図。

【図23】本発明の第7の実施形態に係るプログラム合成装置の概略構成を示すブロック図。

【図24】本発明の第8の実施形態に係る、画像前処理プログラムを合成するプログラム合成装置の具体的な構成を示すブロック図。

【図25】概略処理手順入力部に記憶された概略処理の記憶例を示す図。

【図26】合成された画像前処理プログラムの処理対象となる画像の一例を示す模式図。

【図27】概略処理手順を指定するインタフェース画面の表示例を示す模式図。

【図28】プログラム部品記憶手段における部品分類および各分類に属するプログラム部品の記憶例を示す図。

【図29】プログラム部品記憶手段に記憶されるプログラム部品（プログラムコードと属性情報）のデータ構造例を示す図。

【図30】コーディング情報が表現している各概略処理手順と、染色体ビット列との対応関係の例を示す図。

【図31】図27のウィンドウに表示されている概略処理手順と、図28に示したような部品ファイルに収められているプログラム部品の情報から生成されるコーディング情報の具体例を示す図。

【図32】交叉・突然変異操作部により交叉操作を施す前の2つの染色体A、Bの具体例を示す図。

【図33】2つの染色体A、Bに対し交叉操作を施して得られた染色体A'、B'の具体例を示す図。

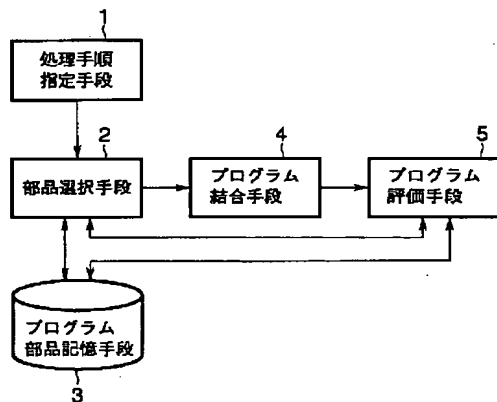
【図34】突然変異操作を施す前の染色体の具体例を示す図。

【図35】突然変異操作を施して得られた染色体の具体例を示す図。

【符号の説明】

- 1…処理手順指定手段
- 2…部品選択手段
- 3…プログラム部品記憶手段
- 4…プログラム結合手段
- 5…プログラム評価手段

【図1】



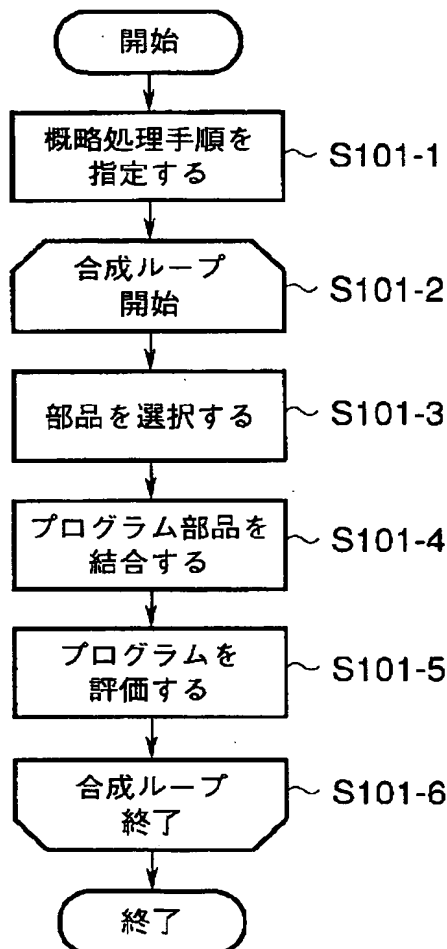
【図7】

プログラム部品	プログラムコード	
	プログラムコードの属性	

【図14】

入力番号	期待されている出力	出力
1	0exp(1)	01
2	0exp(2) (中略)	02
n	0exp(n)	0n

【図2】



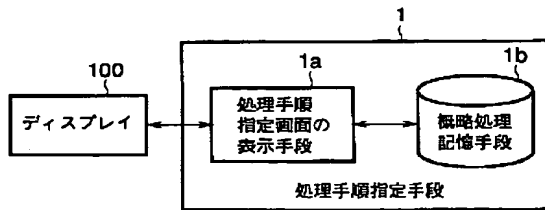
【図4】

概略処理1
概略処理2
...
概略処理3

【図6】

...
部品分類10
プログラム部品10_a
プログラム部品10_b
...
プログラム部品10_c
...
部品分類331
プログラム部品331_a
プログラム部品331_b
...
プログラム部品331_c
...
部品分類510
プログラム部品510_a
プログラム部品510_b
...
プログラム部品510_c
...

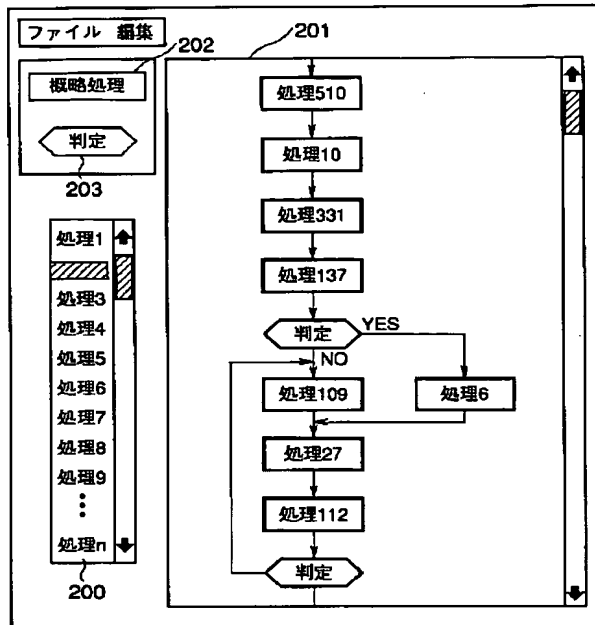
【図3】



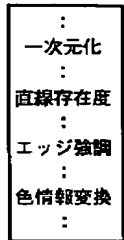
【図15】

プログラム部品	プログラムが保存されているfile名	部品分類
プログラム部品A	file 1	部品分類10
プログラム部品B	file 2	部品分類331
...	...	...
プログラム部品N	file n	部品分類510

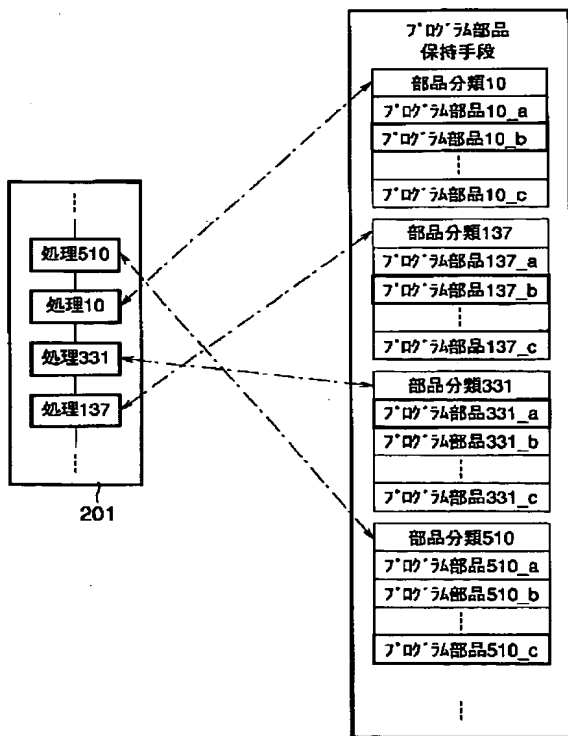
【図5】



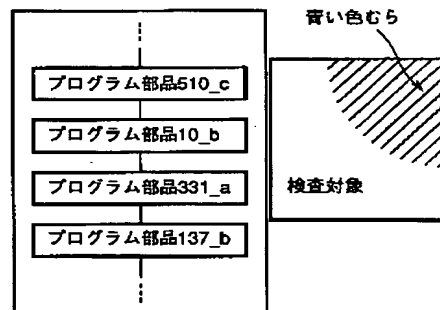
【図25】



【図8】

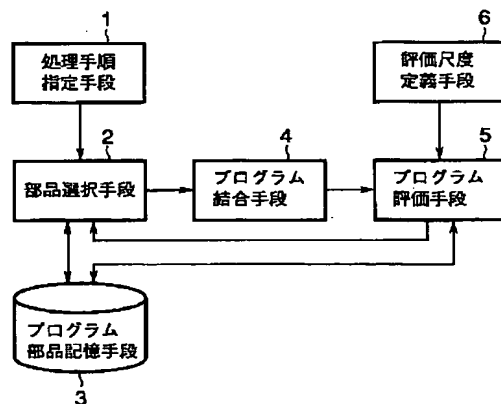


【図9】



【図26】

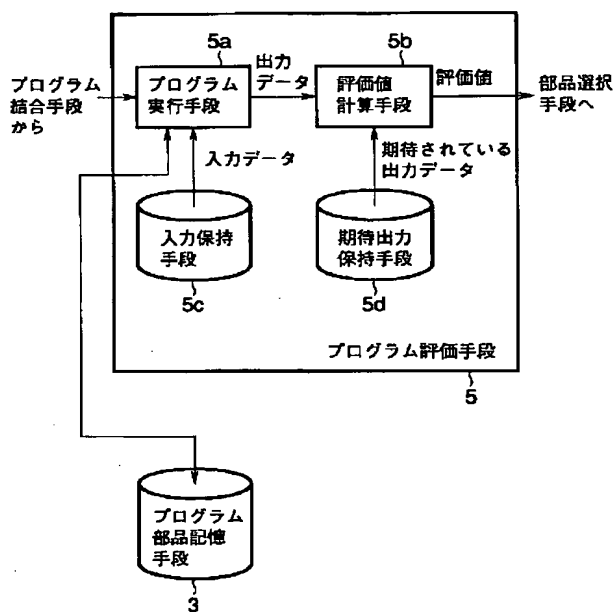
【図16】



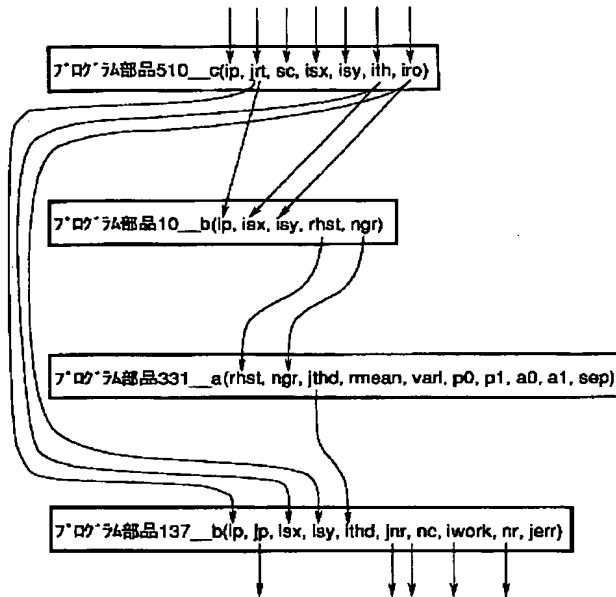
【図10】

プログラム 部品名	属性		
	引数名	入出力種別	内容
プログラム部品10_b	ip	input	Bbbb
	lsx	input	Ffff
	isy	input	Gggg
	rhst	output	Hhhh
	ngr	output (中略)	Iiii
プログラム部品137_b	ip	input	Bbbb
	lp	output	Rrrr
	lsx	input	Ffff
	isy	input	Gggg
	lthd	input	Jjjj
	jnr	output	Ssss
	nc	output	Tttt
	iwork	output	Uuuu
	nr	output	Vvvv
	jerr	output (中略)	Wwww
プログラム部品331_a	rhst	input	Hhhh
	ngr	input	Iiii
	jthd	output	Jjjj
	rmean	output	Kkkk
	var1	output	Llll
	p0	output	Mmmm
	p1	output	Nnnn
	a0	output	Ooooo
	a1	output	Pppp
	sep	output (中略)	Qqqq
プログラム部品510_c	ip	input	Aaaa
	jrt	output	Bbbb
	sc	input	Cccc
	lsx	input	Dddd
	isy	input	Eeee
	lth	in/out	Ffff
	iro	in/out (中略)	Gggg

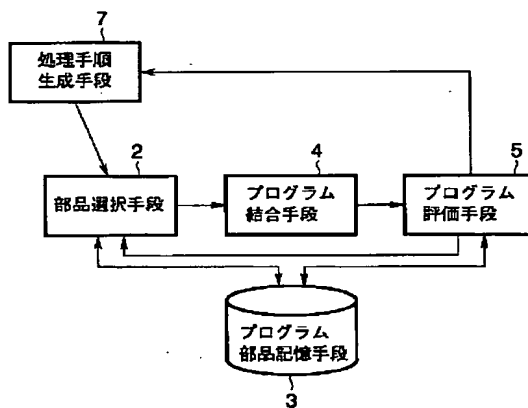
【図13】



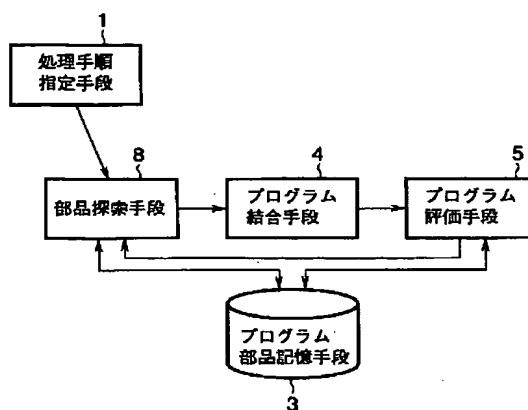
【図12】



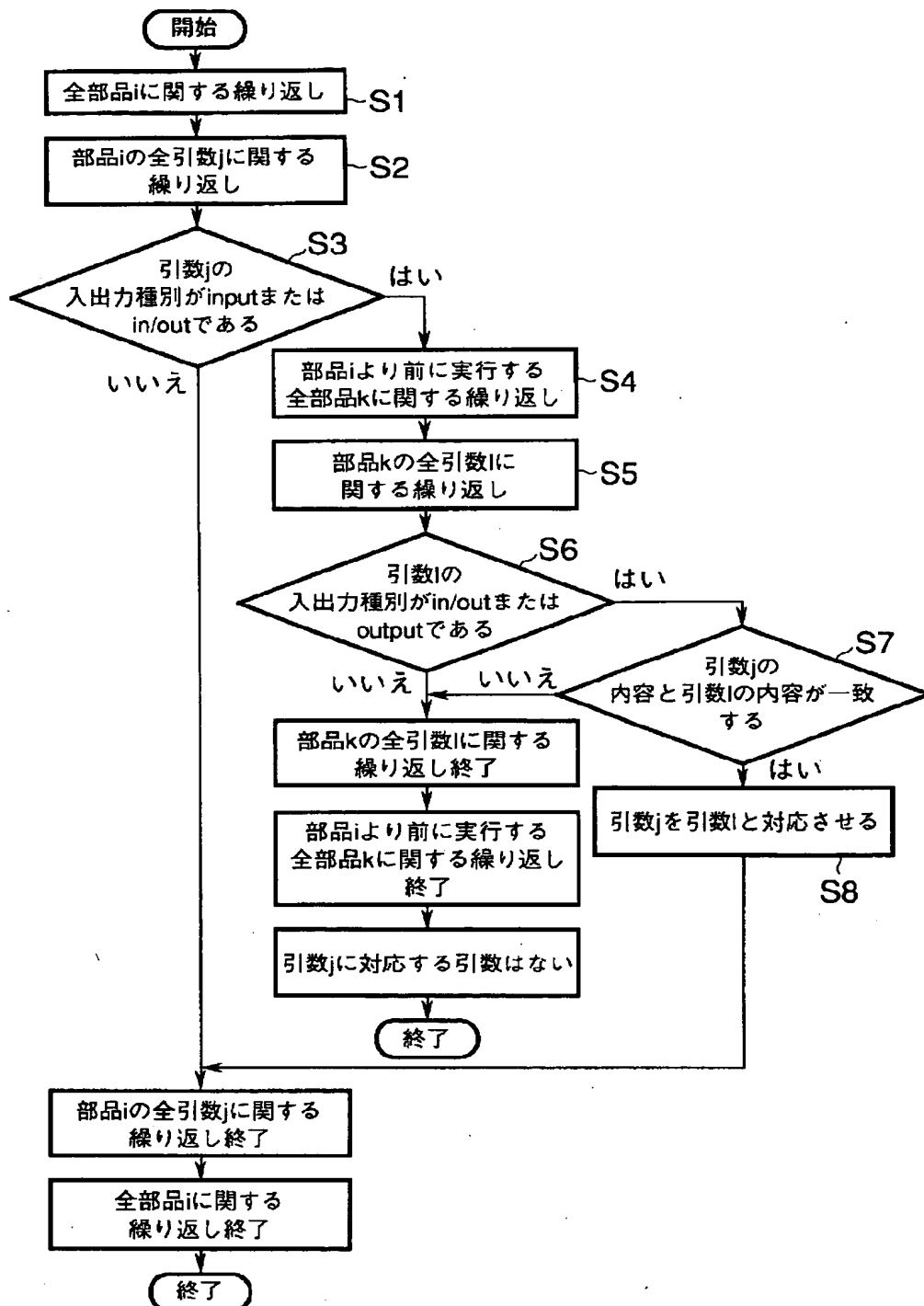
【図18】



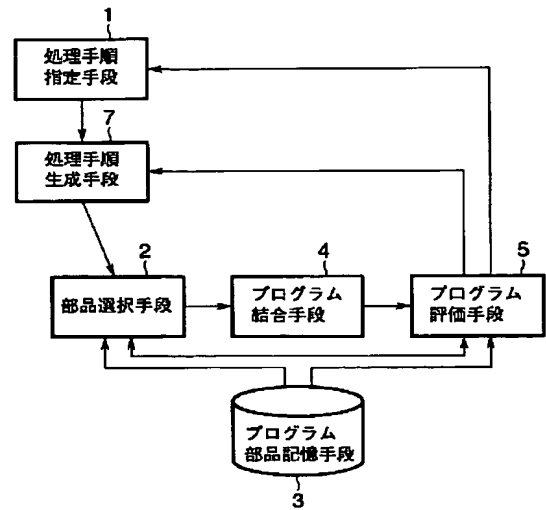
【図23】



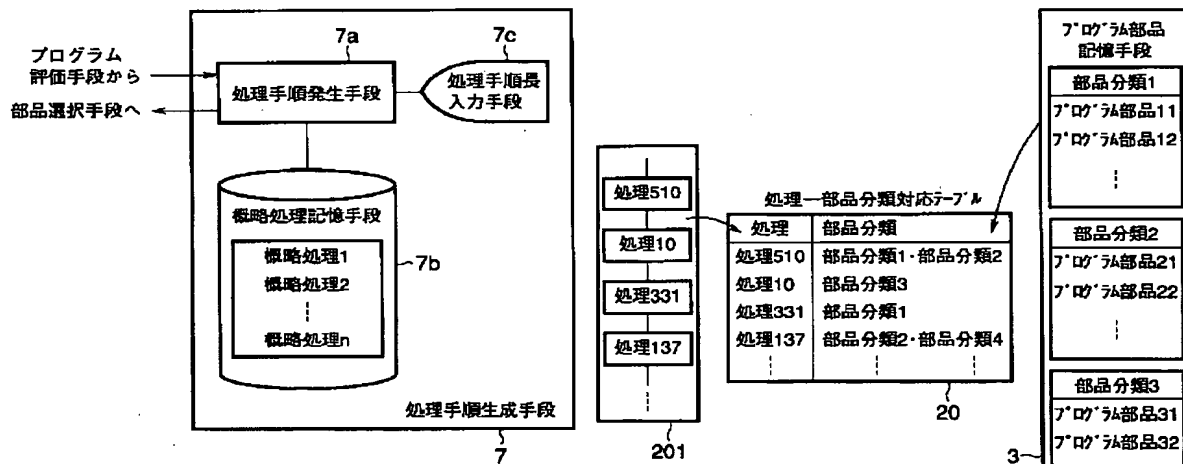
【図11】



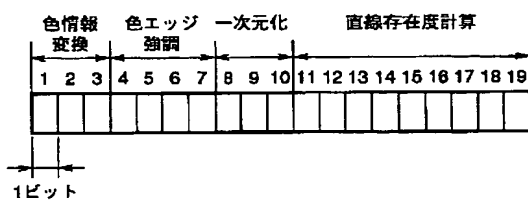
【図 20】



【図 2 1】



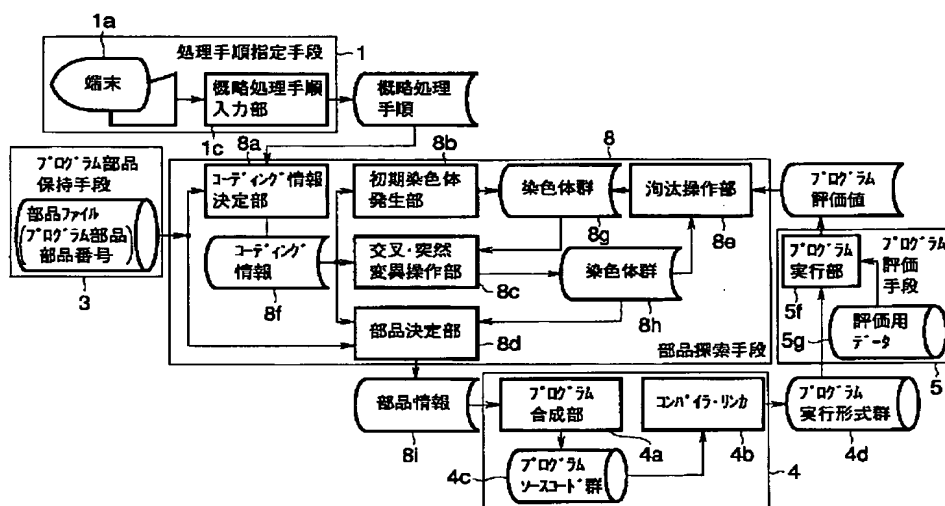
【図 30】



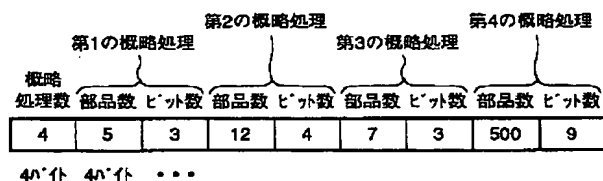
【图 2 2】

7'07"3A部品	属性						
	優先度	ターゲットマシン上での実行時間		ターゲットマシン上での7'07"3Aサイズ		ターゲットマシン上での記憶容量	
		ターゲットマシン1	ターゲットマシン2	ターゲットマシン1	ターゲットマシン2	ターゲットマシン1	ターゲットマシン2
7'07"3A部品1	p <sub>1</sub>	t <sub>11</sub> sec	t <sub>12</sub> sec	s <sub>11</sub> KB	s <sub>12</sub> KB	m <sub>11</sub> KB	m <sub>12</sub> KB
7'07"3A部品2	p <sub>2</sub>	t <sub>21</sub> sec	t <sub>22</sub> sec	s <sub>21</sub> KB	s <sub>22</sub> KB	m <sub>21</sub> KB	m <sub>22</sub> KB
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

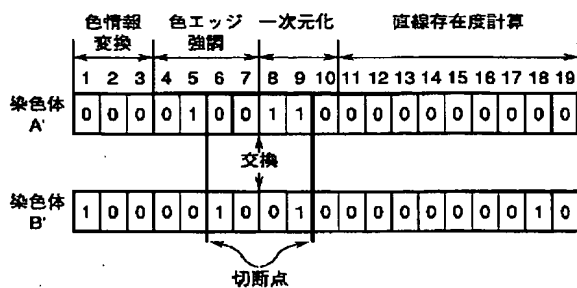
【图 24】



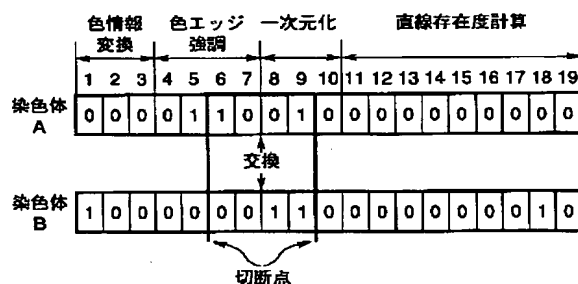
【例 3 1】



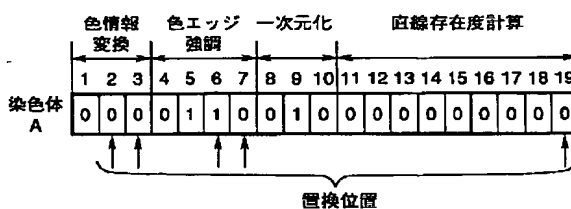
【图 3-3】



【图 3 2】

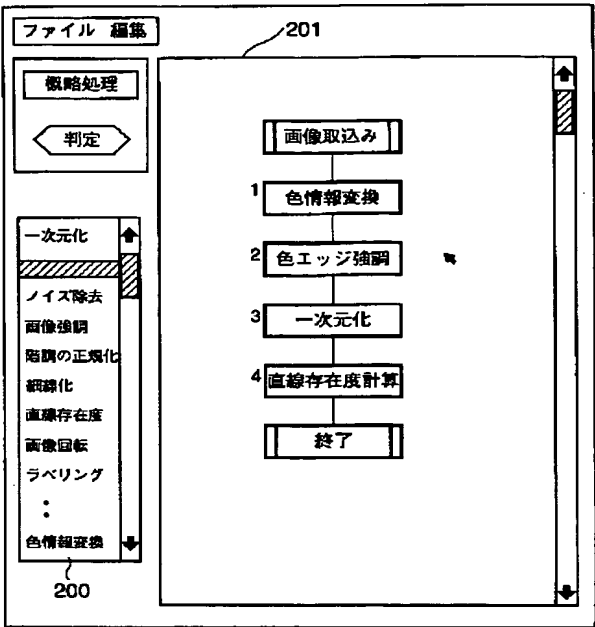


【图 3 4】





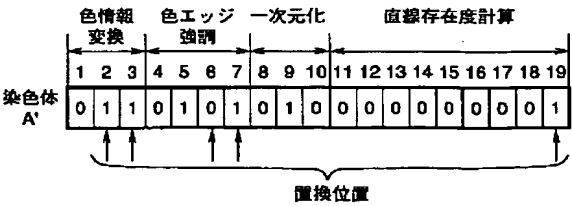
【図 2 7】



【図 2 8】

部品分類	部品番号	プログラム部品
色エッジ強調	0	Prewitt空間フィルタ
	1	Kirsch空間フィルタ3×3
	2	Kirsch空間フィルタ5×5
	3	Robinson空間フィルタ
	4	Sobelフィルタ3×3
	5	Sobelフィルタ5×5
	6	Laplacian空間フィルタタイプ1
	7	Laplacian空間フィルタタイプ2
	8	Laplacian空間フィルタタイプ3
	9	Laplacian空間フィルタタイプ4
	10	Laplacian空間フィルタタイプ5
	11	S-G空間フィルタ
色情報変換	0	色相・彩度変換
	1	色度変換
	2	YIQ変換
	3	VCS変換
	4	無変換
直線存在度計算	0	マスク処理なしハフ変換分割数12
	1	マスク処理なしハフ変換分割数16
	2	マスク処理なしハフ変換分割数20
一次元化	0	$\sqrt{x_1^2 + x_2^2 + x_3^2}$
	1	$\sqrt{x_1^2 + x_2^2}$
	2	$\sqrt{x_2^2 + x_3^2}$
	3	$\sqrt{x_1^2 + x_3^2}$
	4	$x_1$
	5	$x_2$
	6	$x_3$

【図 3 5】



【図 29】

{	
Prewitt空間フィルタ	プログラムコード
	<pre>/* Prewitt filter*/ bool_t prewitt(width, height, input, result) int width; /*画像横幅*/ int height; /*画像縦幅*/ int *input; /*入力画像*/ int *result; /*出力画像*/ {     :</pre>
	プログラム属性
	第一引数 input, 画像横幅 第二引数 input, 画像縦幅 第三引数 input, 入力画像 第四引数 output, 出力画像 4.7秒 2813バイト
{	
色相・彩度変換	プログラムコード
	<pre>/* color &amp; vividress conversion*/ bool_t cvconv(width, height, inputR, inputG, inputB,                resultR, resultG, resultB) int width; /*画像横幅*/ int height; /*画像縦幅*/ int *inputR; /*入力R画像*/ int *inputG; /*入力G画像*/ int *inputB; /*入力B画像*/ int *inputR; /*出力R画像*/ int *inputG; /*出力G画像*/ int *inputB; /*出力B画像*/ {     :</pre>
	プログラム属性
	第一引数 input, 画像横幅 第二引数 input, 画像縦幅 第三引数 input, 入力画像 第四引数 output, 出力画像 2.6秒 1052バイト
{	

フロントページの続き

(72)発明者 後藤 幸博  
神奈川県横浜市磯子区新磯子町33番地 株  
式会社東芝生産技術研究所内

(72)発明者 塚田 弘志  
神奈川県横浜市磯子区新磯子町33番地 株  
式会社東芝生産技術研究所内

(72)発明者 後藤 和久  
神奈川県横浜市磯子区新磯子町33番地 株  
式会社東芝生産技術研究所内

(72)発明者 遠藤 経一  
神奈川県川崎市幸区柳町70番地 株式会  
社東芝柳町工場内

(72)発明者 小野 明  
神奈川県横浜市磯子区新磯子町33番地 株  
式会社東芝生産技術研究所内